An Introduction to Basic Modulation Schemes using Python

Nguyen Van Tinh

Computer Communication Laboratory, The University of Aizu

CCL Winter Camp 2022, Mar 14th

Contents

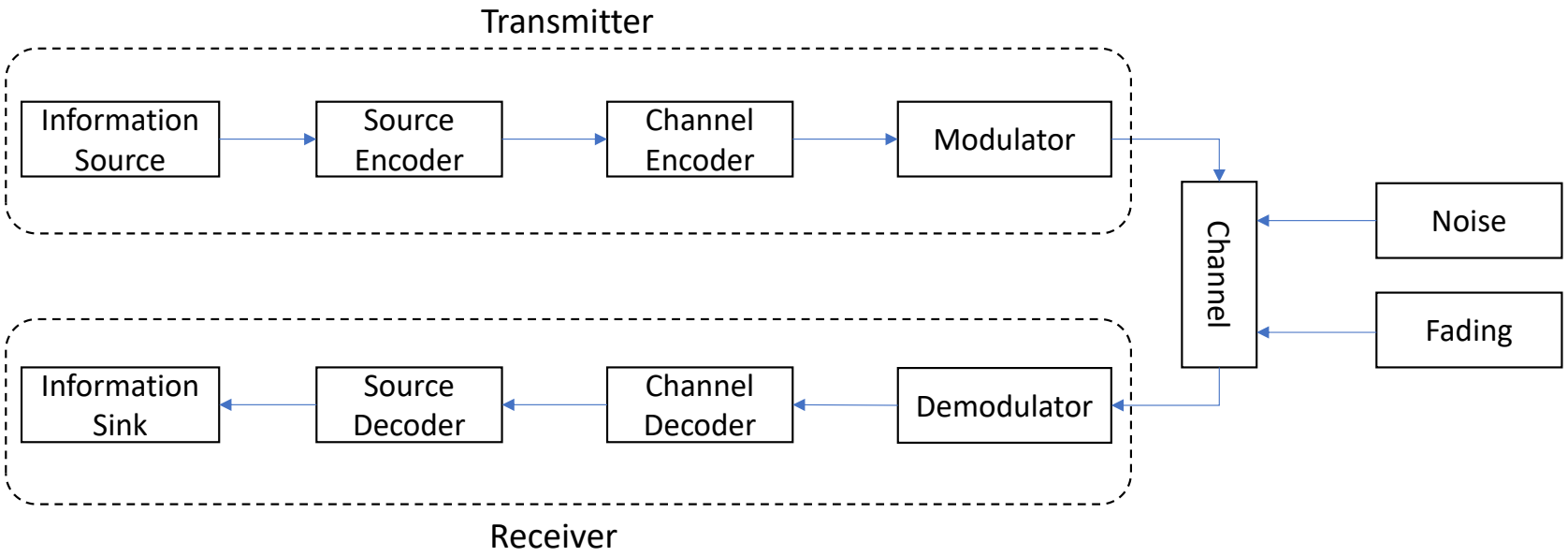
- The Motivation and Background
- A Specific Example – QPSK with Python
- Performance Evaluation
- Conclusions and Future Directions

The Motivation

- Overall goal: applying Machine Learning for a communication issue
  - Current work: Understanding the communication system
    - Simulating to deeply understanding the concept.
    - There are 2 most common tools for simulation:
      - MATLAB: more powerful with a very strong world-wide developer community.
      - Python: becomes more and more popular
  - Future direction: applying ML for a communication issue
    - Python is more popular and preferable

→ Using Python for the communication system simulation.

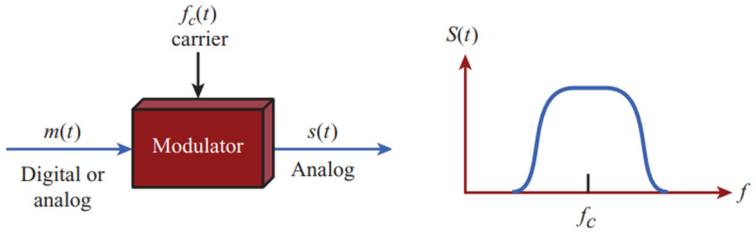
Basic Communication System



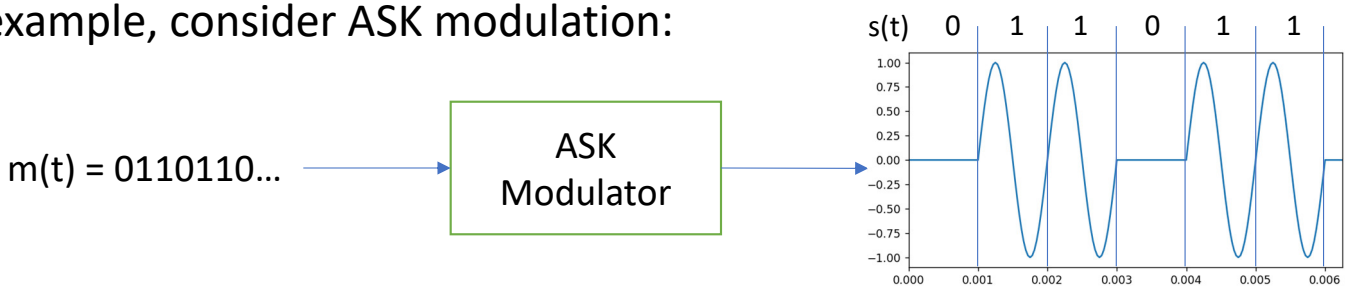
- In this talk, I will only focus on Modulation – Channel – Demodulation parts.
- Consider the performance comparison of different modulation schemes.

Modulation

- Modulation is the process of encoding source data onto a carrier signal with frequency f_C



- A basic signal is expressed as: $s(t) = Acos(2\pi ft + \varphi)$
  - Basic schemes: ASK, BFSK, BPSK → By changing one of three characteristics A, f or φ
  - Multilevel schemes: M-FSK, M-PSK
  - Combined scheme: QAM → A combination of ASK and PSK
- For example, consider ASK modulation:

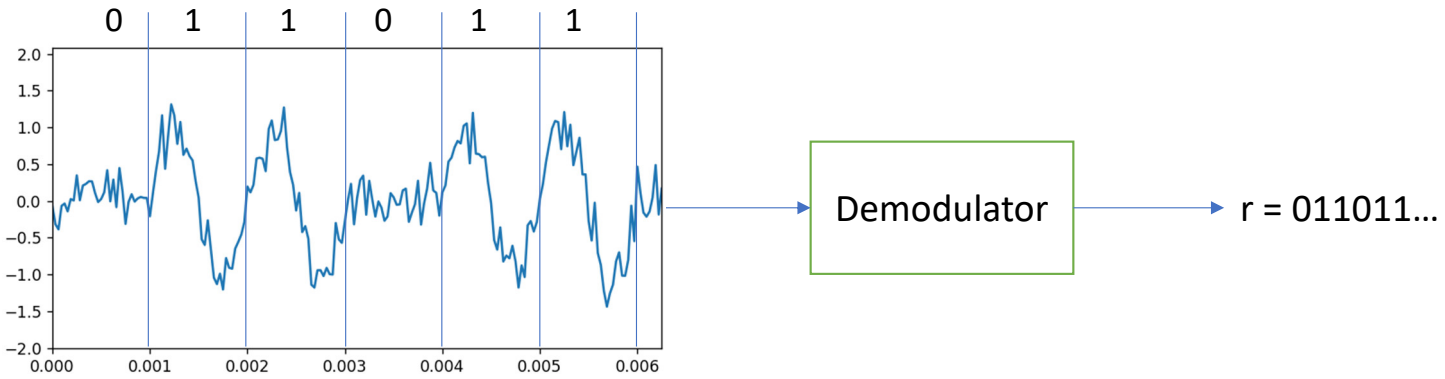


Channel

- Channel: the physical medium that is used to send the signal from the transmitter to the receiver. In wireless communication, the channel may be the atmosphere.
  - Noise: unwanted signal that combines with and distorts the intended signal
    - Thermal noise: present in all electric devices and transmission media; uniformly distributed across the frequency spectrum; also known as white noise.
  - Fading: Refers to the time variation of received signal power caused by changes in the transmission medium or paths.

Demodulation

- Demodulation is a process of extracting the original information-carrying signal from a modulated carrier wave.



Quadrature phase shift keying (QPSK) (1)

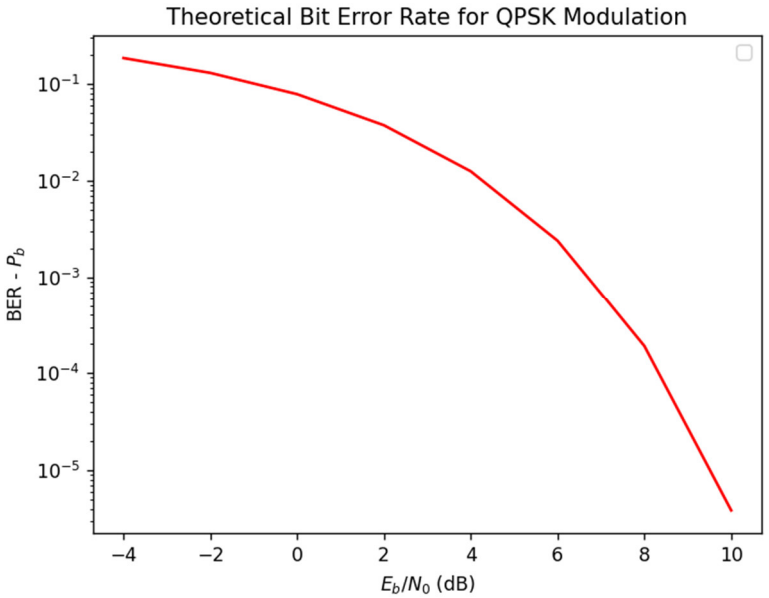
- QPSK is a form of phase modulation technique, in which two bits (combined as one symbol) are modulated at once. The QPSK signal within a symbol duration Tsym is defined as: $s(t) = Acos(2\pi f_c t + \theta_n),\ \mathrm{n} = 1,2,3,4$

Where the signal phase is given by:

$$\theta_n = \frac{(2n-1)\pi}{4}$$

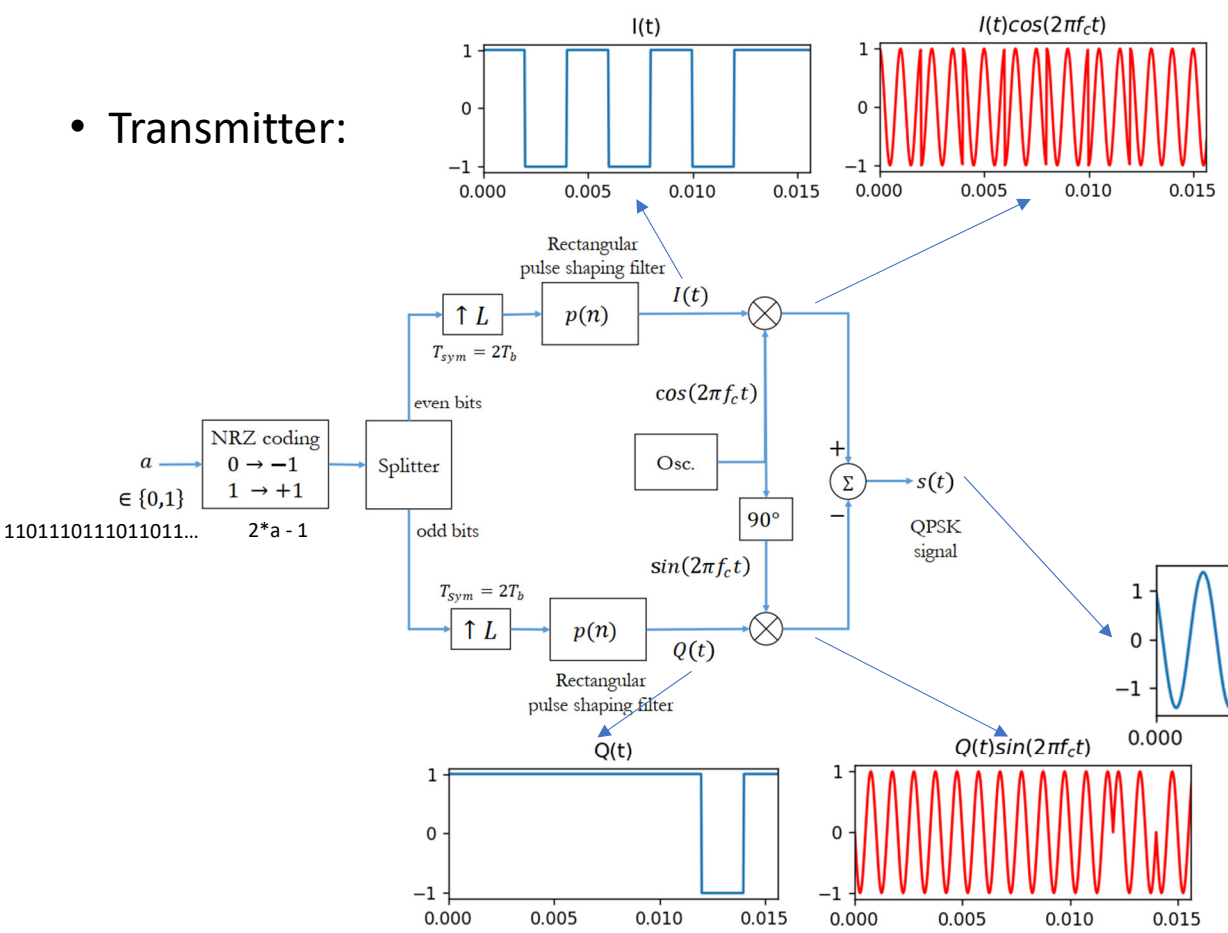
- The QPSK signal can be re-written as:
  $s(t) = Acos\theta_n \cos(2\pi f_c t) - Asin\theta_n \sin(2pi\ f_c t)$
- The theoretical bit error rate of BPSK under additive white Gaussian noise (AWGN) can be calculated as:

$$P_b = \frac{1}{2}\mathrm{erfc}\sqrt{\frac{E_b}{N_0}}$$



Quadrature phase shift keying (QPSK) (2)

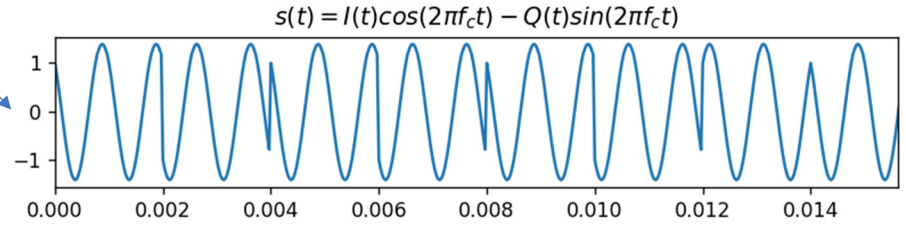
- Transmitter:



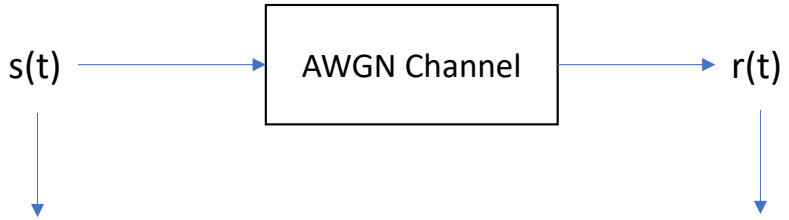

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import upfirdn
from numpy.random import standard_normal
from scipy.special import erfc
```

```
N = 100000  # Num of Symbols to transmit
Fc = 1000   # Carrier freq
OF = 32     # Oversampling factor
L = 2*OF    # samples in each symbol
Fs = OF*Fc  # Sampling Freq
```

```
# generate data to transmit
ak = np.random.randint(2, size=N)
I = a[0::2] # even and odd bit streams
Q = a[1::2]

I = upfirdn(h=[1]*L, x=2*I-1, up=L) # Sampling
Q = upfirdn(h=[1]*L, x=2*Q-1, up=L)
t = np.arange(0,len(I)/fs,1/fs)     #time base

I_t = I*np.cos(2*np.pi*fc*t)
Q_t = -Q*np.sin(2*np.pi*fc*t)
s_t = I_t + Q_t # QPSK modulated baseband signal
```

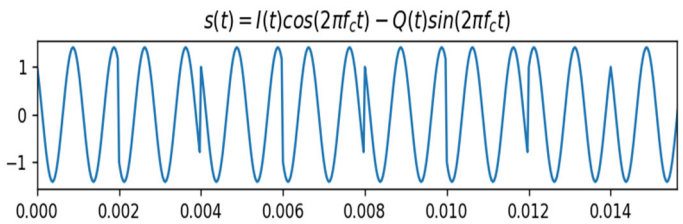


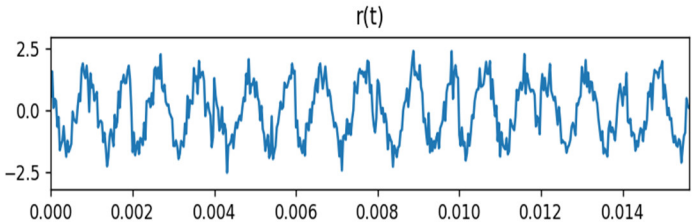
Quadrature phase shift keying (QPSK) (3)




```python
def awgn(s,SNRdB,L=1):
    gamma = 10**(SNRdB/10)      #SNR to linear scale
    P=L*sum(abs(s)**2)/len(s)   #Actual power in the vector
    N0=P/gamma      # Find the noise spectral density
    if isrealobj(s):# check if input is real/complex object type
        n = sqrt(N0/2)*standard_normal(s.shape) # computed noise
    else:
        n = sqrt(N0/2)*(standard_normal(s.shape)+1j*standard_normal(s.shape))

    r = s + n # received signal
    return r
```

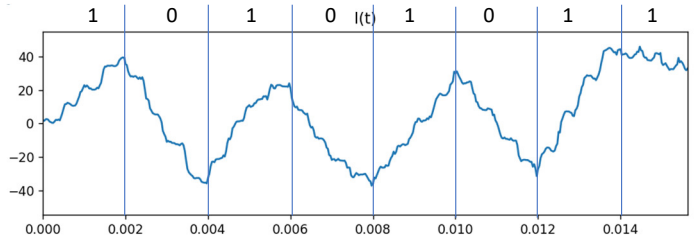
```python
EbN0 = 20    # (Eb/N0) dB
r_t = awgn(s_t, EbN0, L)          # Through AWGN channel
```

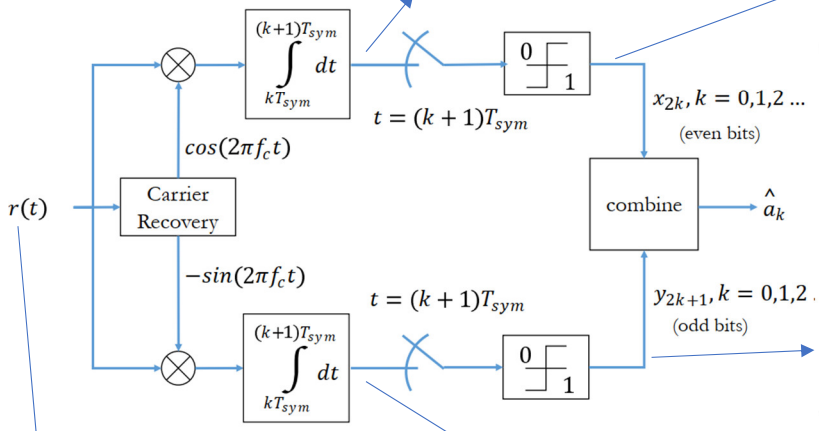


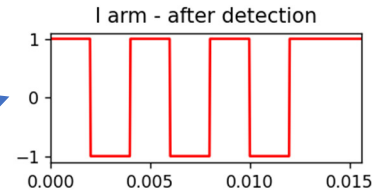


Quadrature phase shift keying (QPSK) (4)

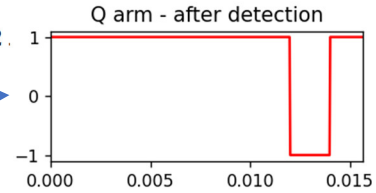
- Receiver:

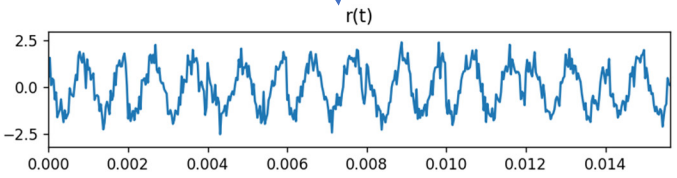


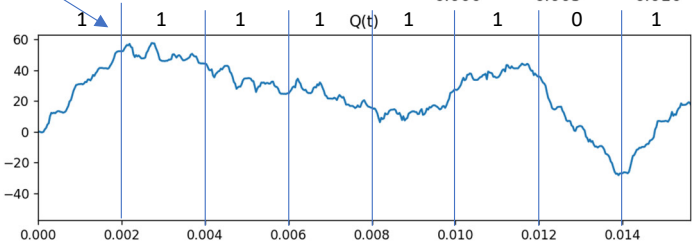




1101110111011011...



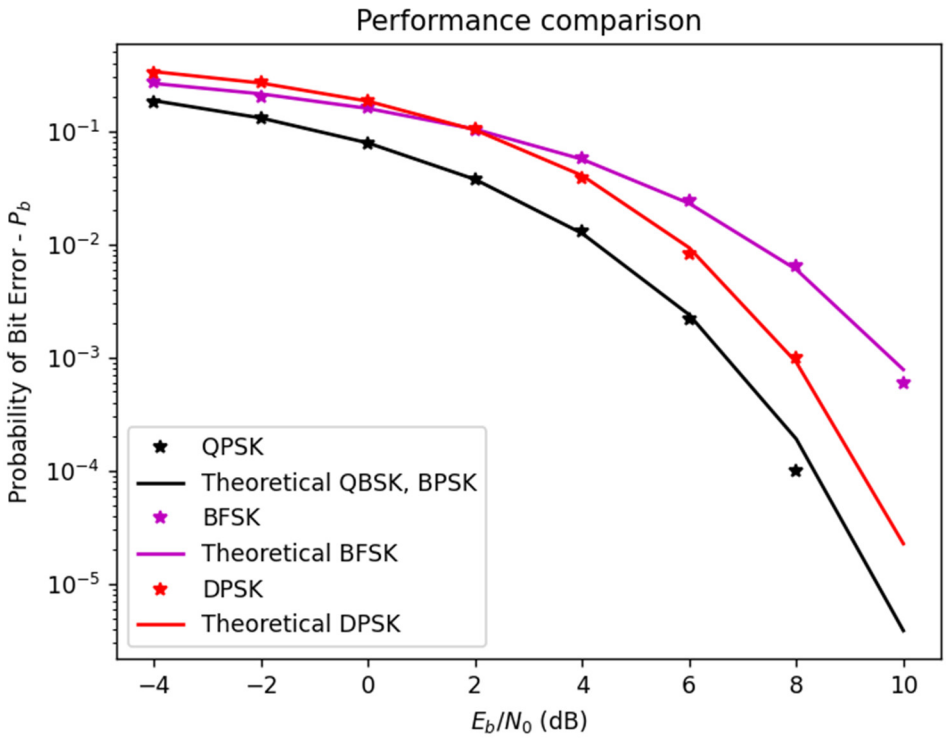





```
x = r_t*np.cos(2*np.pi*fc*t)     # I
y = -r_t*np.sin(2*np.pi*fc*t)    # Q
x = np.convolve(x,np.ones(L))    # integrate for Tsym=2*Tb duration
y = np.convolve(y,np.ones(L))    # integrate for Tsym=2*Tb duration

I = x[L-1::L]  # I arm - sample at every symbol instant Tsym
Q = y[L-1::L]  # Q arm - sample at every symbol instant Tsym

I = I > 0
Q = Q > 0

a_hat = np.zeros(2*len(I))
a_hat[0::2] = I     # even bits
a_hat[1::2] = Q     # odd bits
```

Performance Comparison

- BPSK provides the best performance.
- To achieve the same BER, as compared to BPSK:
  - DPSK needs approximately 1dB more of Eb/N0
  - BFSK needs more approximately 3dB of Eb/N0



Machine Learning for CSI prediction

- Channel state information (CSI) helps wireless systems adapt their transmission parameters to instantaneous channel conditions, which is crucial for achieving a reliable communication with high data rates.
- CSI needs to be estimated at the receiver and usually quantized and feedback to the transmitter.
- CSI is affected by noise, fading, attenuation… as well as tends to be outdated due to high feedback delay.
- My future work: applying Machine Learning to provide a channel predictor to get a perfect CSI.

Conclusions

- Showing how to use Python to simulate the QPSK modulation scheme as a specific example.
- Showing the waveforms of the signal during the modulation and demodulation process.
- Comparing the performance among some basic modulation schemes

Thank you for your listening!